EXHIBIT 8

APPENDIX A-23:

Invalidity of U.S. Patent No. 7,519,814 in view of Solaris Zones

E.D. Tex. Case Nos. 2:24-CV-00064-JRG

Exhibit A-23: Invalidity of U.S. Patent No. 7,519,814 in view of Solaris Zones

<u>Reference:</u> Solaris Zones is a software application for implementing operating system-level virtualization technology. Solaris Zones anticipates and/or renders obvious, including under Plaintiff's apparent infringement theory,[1] all asserted claims[2] of U.S. Patent No. 7,519,814 (the "'814 patent") under 35 U.S.C. §§ 102 and/or 103. Solaris Zones was available at least by February 2004.[3] However, testimony from VirtaMove inventor Don Rochette indicates that a version of Solaris Zones disclosing the asserted claim limitations was available as early as 2002. *E.g.*, Rochette Dep. Tr. 26:3-8 ("Q. And you were aware of an operating system from Sun Microsystems called Solaris? A. Yes. Q. And Solaris predated the two patents that we're looking at; correct? A. That's correct."); 26:13-16; 26:17-21; 54-57. This earlier date for Solaris Zones is further supported by the Tucker references described in Charts A-1, A-2, and A-3. Solaris Zones is prior art to the '058 patent under at least 35 U.S.C. § 102(a).

The chart below provides representative examples of where each element of each claim is found within Solaris Zones, including under Plaintiff's apparent construction of the asserted claims (and to the extent the claims are not found indefinite under 35 U.S.C. § 112). The cited evidence is merely illustrative, and Defendant reserves the right to cite alternative or additional evidence. To the extent Plaintiff contends that Solaris Zones does not disclose any asserted claims or claim elements of the '814 Patent, it would have been obvious to combine the teachings of Solaris Zones with: (1) the knowledge of one of ordinary skill in the art to show all the limitations of the claims; (2) the teachings of any of the prior art references set forth in Defendant's other invalidity charts with respect to the one or more limitations; and/or (3) the teachings of any of the prior art references set forth in the cover document of Defendant's Invalidity Contentions with respect to the '814 patent. Plaintiff has yet to identify any limitation of the asserted claims that it contends is not fully disclosed by Solaris Zones, either alone or in combination with other prior art cited by Defendant and/or with the knowledge

[1] To the extent that these Invalidity Contentions rely on or otherwise embody particular constructions of terms or phrases in the Asserted Claims, Defendant is not proposing any such constructions as proper constructions of those terms or phrases. Various positions put forth in this document are predicated on Plaintiff's incorrect and overly broad interpretation of its claims as evidenced by its Infringement Contentions provided to Defendant. Those positions are not intended to and do not necessarily reflect Defendant's interpretation of the true and proper scope of Plaintiff's claims, and Defendant reserves the right to adopt claim construction positions that differ from or even conflict with various positions put forth in this document.

[2] As used herein, "asserted claims" refers only to those claims charted in Plaintiff's June 5, 2024, Infringement Contentions. To the extent Plaintiff later obtains leave to assert any additional claims for this patent, Defendant will provide its preliminary invalidity contentions consistent with the timing requirements set forth in the Court's order.

[3] Solaris Containers, https://en.wikipedia.org/wiki/Solaris_Containers.

of one of ordinary skill in the art. To the extent Defendant makes any such contention in the future, Defendant expressly reserves the right to rebut any such contention, including by identifying additional obviousness combinations.

Where the chart below states that Solaris Zones "discloses" a limitation, such disclosure may be express or inherent. All emphasis is added unless otherwise indicated.

Fact Discovery is ongoing and Defendant's prior art investigation, including via third party discovery, is therefore not yet complete. Defendant reserves the right to rely upon additional evidence of invalidity obtained in the future as to Solaris Zones or any other prior art public use/sales/offers for sale that may anticipate or render obvious one or more asserted claims of the Asserted Patents under 35 U.S.C. § 102(e) and/or 35 U.S.C. § 103. Defendant may rely on additional documents and/or things that have not yet been located and/or testimony to support the contentions regarding Solaris Zones set forth in this chart.

Invalidity Contentions: Solaris Zones

Claim 1

’814 Patent Claim 1	Disclosure
1[p][1] In a system having a plurality of servers with operating systems that differ, operating in disparate computing environments,	Solaris Zones, as evidenced by the example citations below, discloses a system having a plurality of servers with operating systems that differ, operating in disparate computing environments. *See, e.g.*: • “Solaris has this wonderful new feature called "Zones". In addition to the many features that Solaris usually ships with we now have a new and very powerful feature that allows the administrator to create a virtual server within the server. Your server is no longer just a collection of user accounts and applications but it is now a collection of virtual servers within a global zone server. BSD Jail you say? A chroot environment? The comparison can be made to these older tools in much the same way that you can compare an aardvark to an elephant. They both walk and they both have four legs but one of them will carry you and a ton of cargo, the other, well, it just happens to have one vowel.” https://web.archive.org/web/20060613080532/http:/www.blastwave.org/articles/DMC-0002/index.html. • Rochette Tr. at 50:19-51:5 ("Q. Okay. So by 2002 did software developers in Silicon Valley know that large numbers of computers running different operating systems could be connected in a network?" A. Yes, there were many examples of that scenario you just described by that timeframe.") • Rochette Tr. at 51:7-22 ("Q. And by 2002 did software developers in Silicon Valley know that servers running different operating systems could work together to provide a service? A. Yes, there are also examples of that by that date . . . We discussed things with Chase Bank in New York, and they were using Solaris and Windows together to form various applications, Windows being the front end, Solaris being the data management in the back end, together forming a service."); see also Rochette Tr. at 70:17-73:18. • Rochette Tr. at 75:8-11 ("Q. So Solaris zones, based on your experience, can operate in disparate computing environments? A. Yes.") • Rochette Tr. at 75:15-76:15 ("Q. And Solaris zones was capable [of] running [] on a plurality of servers, meaning multiple servers? A. Yes. Q. And those servers would include a processor; right? A. Yes. Q. And an operating system; right? A. Yes. Q. And a kernel; right? A. Yes. Q. And local system files compatible with the processor? A. Yes. Q. And a server could provide executable applications related to a service? A. Yes. Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server

’814 Patent Claim 1	**Disclosure**
	consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][1], Chart A-2 (Tucker ‘556) at limitation 1[p][1], and Chart A-3 (Tucker Provisional) at limitation 1[p][1] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[p][2] wherein each server includes a processor and an operating system including a kernel a set of associated local system files compatible with the processor,	Solaris Zones, as evidenced by the example citations below, discloses wherein each server includes a processor and an operating system including a kernel a set of associated local system files compatible with the processor. *See, e.g.*: • “The first step here is to get a piece of hardware to test with. You can use an Intel or AMD Opteron unit or a UltraSparc server. For my testing and playing I choose one of many Netra T1 UltraSparc units that I have in a rack. I download the CDROM ISO files from Sun. I then use lofiadm and a few other steps to create a jumpstart server for network booting. Simply put, I did the entire process remotely, from home, with a laptop and a modem. Like I said, Solaris is really slick” https://web.archive.org/web/20060613080532/http:/www.blastwave.org/articles/DMC-0002/index.html. • Rochette Tr. at 38:3-41:6 ("Q. So what are some of the differences between what Trigence AE was doing and what Solaris zones was doing? A. With zones, Solaris would utilize some operating system features to create a separate file area. They had modified the kernel, the internals of Solaris, to support this. It relied on kernel internals to provide separations between applications' end zones. It would provide separate scheduling types of mechanisms and so forth also implemented in the kernel. The Trigence solution made no changes to the kernel -- or didn't require any changes to the kernel. It was all done with what was called function overlays . . . It had nothing to do with separate zones or kernel changes or anything. Q. Okay. So when you say that Solaris zones was different from what Trigence was doing, you were contrasting Solaris zones with the Trigence product, not the patents; correct? A. Yes. Q. So can you

’814 Patent Claim 1	Disclosure
	explain these function overlays to me a little bit more. A. Function overlays use an operating system capability called library preload . . . So if the application called a function called fu, let's say, for example, if fu were existent in a system library and also in a Trigence library, because of the preload, the loader would resolve fu to the Trigence library instead of the system library because the Trigence library was loaded first due to LD_Preload/AppInit_DLL. Q. Okay. So you mentioned that this function overlay capability that Trigence was using was called LD_Preload in Solaris; is that right? A. Correct. Q. Okay. So Solaris had this LD_Preload capability that could be used to provide the function overlay capability? A. Correct. Q. And do you know when Solaris gained that function overlay capability that Trigence was using. A. That capability has been present in Solaris since its first release of its origin. Q. Does the 1987 sounds like approximately the right timeframe? A. Yeah, that would be -- that would be in the range, yes.") • Rochette Tr. at 59:10-16 ("Q. By 2002 did software developers in Silicon Valley know that containers could run on top of a kernel that was residing outside of the caper? A. Yes.") • Rochette Tr. at 75:15-76:15 ("Q. And Solaris zones was capable [of] running [] on a plurality of servers, meaning multiple servers? A. Yes. Q. And those servers would include a processor; right? A. Yes. Q. And an operating system; right? A. Yes. Q. And a kernel; right? A. Yes. Q. And local system files compatible with the processor? A. Yes. Q. And a server could provide executable applications related to a service? A. Yes. Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from

’814 Patent Claim 1	Disclosure
	accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") • “These efforts differ from virtual machine implementations in that there is only one underlying operating system kernel, which is enhanced to provide increased isolation between groups of processes. The result is the ability to run multiple applications in isolation from each other within a single operating system instance.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][2], Chart A-2 (Tucker ‘556) at limitation 1[p][2], and Chart A-3 (Tucker Provisional) at limitation 1[p][2] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[p][3] a method of providing at least some of the servers in the system with secure, executable, applications related to a service,	Solaris Zones, as evidenced by the example citations below, discloses a method of providing at least some of the servers in the system with secure, executable, applications related to a service. *See, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”)

Invalidity Contentions: Solaris Zones

’814 Patent Claim 1	Disclosure
	• Rochette Tr. at 76:13-15 ("Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") • Rochette Tr. at 75:15-76:15 ("Q. And Solaris zones was capable [of] running [] on a plurality of servers, meaning multiple servers? A. Yes. Q. And those servers would include a processor; right? A. Yes. Q. And an operating system; right? A. Yes. Q. And a kernel; right? A. Yes. Q. And local system files compatible with the processor? A. Yes. Q. And a server could provide executable applications related to a service? A. Yes. Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][3], Chart A-2 (Tucker ‘556) at limitation 1[p][3], and Chart A-3 (Tucker Provisional) at limitation 1[p][3] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[p][4] wherein the applications are executed in a secure environment,	Solaris Zones, as evidenced by the example citations below, discloses wherein the applications are executed in a secure environment, as this claim limitation appears to be interpreted by VirtaMove. *See, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 76:13-15 ("Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") • Rochette Tr. at 75:15-76:15 ("Q. And Solaris zones was capable [of] running [] on a plurality of servers, meaning multiple servers? A. Yes. Q. And those servers would include a processor; right? A. Yes. Q. And an operating system; right? A. Yes. Q. And a kernel; right? A. Yes. Q. And local system files compatible with the processor? A. Yes. Q. And a server could provide executable applications related to a service? A. Yes. Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.")

’814 Patent Claim 1	Disclosure
	• Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][4], Chart A-2 (Tucker ‘556) at limitation 1[p][4], and Chart A-3 (Tucker Provisional) at limitation 1[p][4] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the

Invalidity Contentions: Solaris Zones

’814 Patent Claim 1	Disclosure
	general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[p][5] wherein the applications each include an object executable by at least some of the different operating systems for performing a task related to the service, the method comprising	Solaris Zones, as evidenced by the example citations below, discloses wherein the applications each include an object executable by at least some of the different operating systems for performing a task related to the service, the method comprising. *See, e.g.*: • Rochette Tr. at 75:15-76:15 ("Q. And Solaris zones was capable [of] running [] on a plurality of servers, meaning multiple servers? A. Yes. Q. And those servers would include a processor; right? A. Yes. Q. And an operating system; right? A. Yes. Q. And a kernel; right? A. Yes. Q. And local system files compatible with the processor? A. Yes. Q. And a server could provide executable applications related to a service? A. Yes. Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files

’814 Patent Claim 1	Disclosure
	that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[p][5], Chart A-2 (Tucker ‘556) at limitation 1[p][5], and Chart A-3 (Tucker Provisional) at limitation 1[p][5] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[a][1] storing in memory accessible to at least some of the servers	Solaris Zones, as evidenced by the example citations below, discloses storing in memory accessible to at least some of the servers. *See, e.g.*: • “A zone can either be bound to a dedicated pool of resources (such as a number of CPUs or a quantity of physical memory), or can share resources with other zones according to defined proportions.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the

’814 Patent Claim 1	Disclosure
	container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[a][1], Chart A-2 (Tucker ‘556) at limitation 1[a][1], and Chart A-3 (Tucker Provisional) at limitation 1[a][1] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[a][2] a plurality of secure containers of application software, each container comprising one or more of the executable applications and a set of associated system files required to execute the one or more applications, for use with a local kernel residing permanently on one of the servers;	Solaris Zones, as evidenced by the example citations below, discloses a plurality of secure containers of application software, each container comprising one or more of the executable applications and a set of associated system files required to execute the one or more applications, for use with a local kernel residing permanently on one of the servers. *See, e.g.*:

’814 Patent Claim 1	Disclosure
	• Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 58:8-20 (“Q. And are you familiar with something called BSD jails? A. Yes. I forgot about those. Yes, I am. Q. And what was BSD jails? A. It was an open source response to the capabilities in Solaris zones. It was a way of obtaining very similar behavior exemplified by zones in an open source Unix capability that didn’t require licensing yet. They’re not exactly the same, but it was close enough. Q. So is BSD jails another container capability? A. Yes.”) • Rochette Tr. at 76:13-15 ("Q. And Solaris zones allowed applications to be executed in a secure environment? A. Correct.") • Rochette Tr. at 52:6-11 ("Q. By 2002 did software developers in Silicon Valley know that software could run inside of containers? A. Yes. . . . Q. Did software developers in Silicon Valley by 2002 know that servers could host more than one container? A. Yes.") • Rochette Tr. at 59:10-16 ("Q. By 2002 did software developers in Silicon Valley know that containers could run on top of a kernel that was residing outside of the caper? A. Yes.") • Rochette Tr. at 75:8-11 ("Q. So Solaris zones, based on your experience, can operate in disparate computing environments? A. Yes.") • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to

’814 Patent Claim 1	Disclosure
	have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “Zones build on this concept by extending this virtual operating system environment to include many of the features of a separate machine, such as a perzone console, system log, packaging database, run level, identity (including name services), and interprocess communication facility. For example, each zone has a virtualized view of the process table (as reflected in the /proc file system) that reflects only the processes running in that zone, as well as a virtualized/etc/mnttab file that shows only file system mounts within the zone.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique

’814 Patent Claim 1	Disclosure
	balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[a][2], Chart A-2 (Tucker ‘556) at limitation 1[a][2], and Chart A-3 (Tucker Provisional) at limitation 1[a][2] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[a][3] wherein the set of associated system files are compatible with a local kernel of at least some of the plurality of different operating systems, the containers of application software excluding a kernel,	Solaris Zones, as evidenced by the example citations below, discloses wherein the set of associated system files are compatible with a local kernel of at least some of the plurality of different operating systems, the containers of application software excluding a kernel. *See, e.g.*: • Rochette Tr. at 27:3-20 (“Q. And are you aware of any examples, from your personal experience, of developers moving the functionality from an operating system’s kernel to a shared library? . . . A. A feature in the Sun Microsystems operating system called Solaris, the feature is called zones.”) • Rochette Tr. at 83:22-88:8 ("Q. I think you mentioned earlier that Solaris zones had the capability of moving certain functionality that would normally be in an operating system kernel into a shared library; is that right? A. Yes, there were some capabilities that were represented outside of the kernel in user mode, correct. Q. What capabilities were those? A. File system separation. The use of something like the chroot command, or the chroot command you referred to earlier. The -- oh, I can't think of other examples. I mean, there were several examples of it. Q. And how did you learn that Solaris zones had that capability? A. Sitting down in front of a Solaris operating system and using it. Reading the documentations and using the capabilities and testing it. Q. And how did you determine that those capabilities were implemented in shared libraries rather than the kernel? A. If you go to an application and -- you would execute the command LDD. LDD shows you what libraries are required by a certain executable, and it would -- it's very clear -- clearly shown what libraries are referenced, and you could then look at the libraries and very clearly see what is implemented in those libraries. Q. So do you know if there was any shared library functionality in Solaris that was also in the kernel, in other words, the shared library functionality was a

’814 Patent Claim 1	Disclosure
	duplicate or performing a similar function to what was in the kernel? A. There were some networking capabilities and shared libraries and some -- for example, that would define what IP address is used, define what routing is used for a container separate from the kernel and was -- would be accessed by applications before going to the kernel. So you would have had a very similar capability in the kernel that was accessed by applications in these zones before going to the kernel. Q. So just for the sake of a concrete discussion, can you think of an example of a shared library that we could talk about that had this operating system functionality? A. I can't recall the names of them in Solaris. They would have been a lib.network.so or lib -- something with the name 'network' in it or 'net.' And it would have the network -- the things that would otherwise -- the things that were also present in a kernel, like defining an IP address, like defining routes, like defining -- not TCP protocol, but these would have been used by zones to configure the network capabilities present in that zone or container for use by the application. I don't remember the names. It's a long time ago, but there were shared libraries used by Solaris that did those things. Q. So for the networking shared library, if there were two applications using that shared library, would they each have their own instance in the sense of, you know, internal data used by the shared library having two different copies of that data? A. The way it works is the code for that library in question, once loaded into memory, is shared by all applications. That's the -- hence, the term 'shared library.' And -- so multiple zones could make use of the same networking library. The difference is that the data they use is distinct. So the actual code that gets executed is common across zones; the data that they use is separate for each zone. And any writes that would happen into memory . . . would implement an operating feature called copy-on-write, which would create a different instance of a page in memory, but it's specific to that zone.") • Rochette Tr. at 59:10-16 ("Q. By 2002 did software developers in Silicon Valley know that containers could run on top of a kernel that was residing outside of the caper? A. Yes.") Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[a][3], Chart A-2 (Tucker ‘556) at limitation 1[a][3], and Chart A-3 (Tucker Provisional) at limitation 1[a][3] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[a][4] wherein some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server,	Solaris Zones, as evidenced by the example citations below, discloses wherein some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server. *See, e.g.*, the disclosures set forth at claim elements 1[a][1]-[3]. *See also, e.g.*:

’814 Patent Claim 1	Disclosure
	• Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 53:15-21 (“Q. By 2002, did software developers in Silicon Valley know that putting an application in a container could prevent the application from accessing files in another container? A. Yes. In the context of Solaris zones, that is a true statement.”) • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A.

’814 Patent Claim 1	Disclosure
	Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[a][4], Chart A-2 (Tucker ‘556) at limitation 1[a][4], and Chart A-3 (Tucker Provisional) at limitation 1[a][4] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[a][5] wherein said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local system files that remain resident on the server, and	Solaris Zones, as evidenced by the example citations below, discloses wherein said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local system files that remain resident on the server. *See, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 53:15-21 (“Q. By 2002, did software developers in Silicon Valley know that putting an application in a container could prevent the application from accessing files in another container? A. Yes. In the context of Solaris zones, that is a true statement.”) • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure

’814 Patent Claim 1	Disclosure
	environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[a][5], Chart A-2 (Tucker ‘556) at limitation 1[a][5], and Chart A-3 (Tucker Provisional) at limitation 1[a][5] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in

Invalidity Contentions: Solaris Zones

’814 Patent Claim 1	Disclosure
	Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[a][6] wherein the application software cannot be shared between the plurality of secure containers of application software, and	Solaris Zones, as evidenced by the example citations below, discloses wherein the application software cannot be shared between the plurality of secure containers of application software. *See, e.g.*: • Rochette Tr. at 37:1-14 (“Q. Okay. So what was your understanding of Solaris zones? A. Oh gosh. That is a lengthy answer. Zones would provide a separate container, a separate security environment for each app and for our multiple -- and for an application to run it. So you could define a zone for each application you had running in Solaris. That zone would allow the application to have it -- potentially its own version of files. It would -- I don't recall if it had isolated networking, but it would provide a secure environment, and it would provide a separate instance of files and so forth.”) • Rochette Tr. at 53:15-21 (“Q. By 2002, did software developers in Silicon Valley know that putting an application in a container could prevent the application from accessing files in another container? A. Yes. In the context of Solaris zones, that is a true statement.”) • Rochette Tr. at 54:2-12 (“Q. By 2002, did software developers in Silicon Valley know that putting an application in a container could prevent the application from interfering with application in another container? A. Again, in the context of Solaris zones, that would be an accurate statement.”) • Rochette Tr. at 81:4-8 (“Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No.”) • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be

Invalidity Contentions: Solaris Zones

’814 Patent Claim 1	Disclosure
	able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “Server consolidation, which allows multiple workloads to run on the same system, has become increasingly important as a way to improve the utilization of computing resources and reduce costs. Consolidation is common in mainframe environments, where technology to support running multiple workloads and even multiple operating systems on the same hardware has been evolving since the late 1960's. This technology is now becoming an important differentiator in the UNIX and Linux server market as well, both at the low end (virtual web hosting) and high end (traditional data center server consolidation). This paper introduces Solaris Zones (zones), a fully realized solution for server consolidation projects in a commercial UNIX operating system. By creating virtualized application execution environments within a single instance of the operating system, the facility strikes a unique balance between competing requirements. On the one hand, a system with multiple workloads needs to run those workloads in isolation, to ensure that applications can neither observe data from other applications nor affect their operation. It must also prevent applications from over-consuming system resources. On the other hand, the system as a whole has to be flexible, manageable, and observable, in order to reduce administrative costs and increase efficiency. By focusing on the support of multiple application

Invalidity Contentions: Solaris Zones

’814 Patent Claim 1	Disclosure
	environments rather than multiple operating system instances, zones meets isolation requirements without sacrificing manageability.” Price , D. and Tucker, A. (2004) *Solaris Zones: Operating System Support for Consolidating Commercial Workloads.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[a][6], Chart A-2 (Tucker ‘556) at limitation 1[a][6], and Chart A-3 (Tucker Provisional) at limitation 1[a][6] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
1[a][7] and wherein each of the containers has a unique root file system that is different from an operating system's root file system.	Solaris Zones, as evidenced by the example citations below, discloses wherein each of the containers has a unique root file system that is different from an operating system's root file system. *See, e.g.*: • “Currently available utilities like chroot, BSD jails and Solaris zones provide the ability to create multiple file systems that are separated from the underlying OS file system.”; VM_HPE_0000865 (2004 SR&ED Technical Submission) • Rochette Tr. at 56:1-4 ("Q. By 2002 did software developers in Silicon Valley know that containers could have their own root file systems? . . . A. That is an accurate statement in the context of Solaris zones.") • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one

Invalidity Contentions: Solaris Zones

’814 Patent Claim 1	Disclosure
	outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") • “Zones are a new operating system abstraction for partitioning systems, allowing multiple applications to run in isolation from each other on the same physical hardware. This isolation prevents processes running within a zone from monitoring or affecting processes running in other zones, seeing each other’s data, or manipulating the underlying hardware. Zones also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “Since each zone has its own name service identity, it also has its own notion of a password file and its own root user.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* Defendant incorporates Chart A-1 (Tucker ‘080) at limitation 1[a][7], Chart A-2 (Tucker ‘556) at limitation 1[a][7], and Chart A-3 (Tucker Provisional) at limitation 1[a][7] as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.

Invalidity Contentions: Solaris Zones

Claim 2

’814 Patent Claim 2	Disclosure
2. A method as defined in claim 1, wherein each container has an execution file associated therewith for starting the one or more applications.	Solaris Zones, as evidenced by the example citations below, discloses the method as defined in claim 1, wherein each container has an execution file associated therewith for starting the one or more applications. *See, e.g.*: • Rochette Tr. at 76:21-83:21 ("Q. Could a server running Solaris zones provide executable applications relating to a service? A. Yes. Q. And could a server running Solaris zones execute applications in a secure environment? A. Yes. Q. And could a server running Solaris zones store multiple secure containers in memory? A. Yes. Q. And could those multiple containers have applications and system files in them? A. Yes. Q. And would those containers use the kernel from the operating system hosted on the server? A. Yes. Q. And that kernel would be on the server regardless of whether it was running any containers; right? A. Yes. Q. Would the kernel be running on the operating system regardless of whether the server was running any containers? A. Yes. The kernel would run first and foremost, yes, in all instances. Q. And would the system files in the containers be compatible with the kernel? A. That is a major difference. In zones they would have to be compatible with the kernel and the version of the kernel being used, yes. Q. And did each container have its own kernel or did they share a kernel? A. One kernel shared by all applications. Q. And could a container contain system files that were also present outside of the container? . . . Could you have system files, like, the configuration file for Apache, one inside the container and one outside the container? A. Yes . . . That's actually a common scenario. Q. And would the instance of Apache inside the container use the configuration file inside of the container instead of the one outside of the container? A. Yes. Q. And would the applications inside of a container on Solaris zones be able to access software in another container? A. No. Q. And would each of those containers be able to have its own root file system? A. Yes. Q. And do you know if Solaris zones allowed resource limits to be placed on a container? A. Yes, it is. Q. And would applications inside of a container be prevented from accessing system files outside of the container? A. Yes, that is correct. Q. And do you know how a container would be created in Solaris zones, in other words, how you would get the necessary files into the container? A. There were very specific command line options that you used with Solaris zones. A zone_create command and a zone command for specifying resources, a zone command for specifying files that were used in the zone, and things like that. So you would -- you would configure these things using a command line. Q. So did Solaris zones have the ability to create a container – basically take an application that’s already outside of a container and then make a copy of it inside of a container? A. Yes, you would make a copy of it. You would create the zone, then you would copy files as needed, very likely you’d have to modify the configuration of something, and then you could start your application. Q. And when you would copy files into the container, it would be applications and system files, or no? A. Application files for sure. System files would be option, depending on what you were doing. You might need some system files, you might not, depending on what the application was and what you were trying

’814 Patent Claim 2	Disclosure
	to accomplish. Q. And when you made that copy, you would still have the original version outside of the container; right? A. You would -- it would be possible, yes.") Defendant incorporates Chart A-1 (Tucker ‘080) at claim 2, Chart A-2 (Tucker ‘556) at claim 2, and Chart A-3 (Tucker Provisional) at claim 2 as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.

Invalidity Contentions: Solaris Zones

Claim 6

’814 Patent Claim 6	Disclosure
6. A method as defined in claim 2, comprising the step of assigning a unique associated identity to each of a plurality of the containers, wherein the identity includes at least one of IP address, host name, and MAC address.	Solaris Zones, as evidenced by the example citations below, discloses the method as defined in claim 2, comprising the step of assigning a unique associated identity to each of a plurality of the containers, wherein the identity includes at least one of IP address, host name, and MAC address. *See, e.g.*, the disclosures set forth at claim elements 1[p][2]-[4] and 1[a][1]. *See also, e.g.:* • Rochette Tr. at 62:2-18 ("Q. And by 2002 did software developers in Silicon Valley know that containers could have their own IP address separate from a host . . . But would it surprise you if someone had that capability before 2002? A. It would not surprise me -- no.") Defendant incorporates Chart A-1 (Tucker ‘080) at claim 6, Chart A-2 (Tucker ‘556) at claim 6, and Chart A-3 (Tucker Provisional) at claim 6 as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.

Claim 9

’814 Patent Claim 9	Disclosure
9. A method as defined in claim 2, wherein server information related to hardware resource usage including at least one of CPU memory, network bandwidth, and disk allocation is associated with at least some of the containers prior to the applications within the containers being executed.	Solaris Zones, as evidenced by the example citations below, discloses the method as defined in claim 2, wherein server information related to hardware resource usage including at least one of CPU memory, network bandwidth, and disk allocation is associated with at least some of the containers prior to the applications within the containers being executed. *See, e.g.*, the disclosures set forth at claim elements 1[p][2], 1[p][4], 1[a][1]. *See also*: • “The proportion of CPU resources that a zone can consume can be defined by an administator, and then that share can be further divided among workloads running in the zone by the (potentially different) zone administrator. In addition, the privileges available within a zone (even to the root user) are restricted to those that can only affect the zone itself. As a result, even if a zone is compromised by an intruder, the compromise will not affect other zones in then system or the system as a whole.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* • “A zone can either be bound to a dedicated pool of resources (such as a number of CPUs or a quantity of physical memory), or can share resources with other zones according to defined proportions.” Tucker et al., *Solaris Zones: Operating System Support for Server Consolidation.* This element is satisfied by the disclosures set forth at claim elements 1[p][2], 1[p][4], and 1[a][1]. Defendant incorporates Chart A-1 (Tucker ‘080) at claim 9, Chart A-2 (Tucker ‘556) at claim 9, and Chart A-3 (Tucker Provisional) at claim 9 as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.

Invalidity Contentions: Solaris Zones

Claim 10

’814 Patent Claim 10	Disclosure
10. A method as defined in claim 2, wherein in operation when an application residing within a container is executed, said application has no access to system files or applications in other containers or to system files within the operating system during execution thereof.	Solaris Zones, as evidenced by the example citations below, discloses the method as defined in claim 2, wherein in operation when an application residing within a container is executed, said application has no access to system files or applications in other containers or to system files within the operating system during execution thereof. *See*, *e.g.*, the disclosures set forth at claim elements 1[p][2]-[4] and 1[a][1]. This element is satisfied by the disclosures set forth at claim elements 1[p][2]-[4] and 1[a][1]. Defendant incorporates Chart A-1 (Tucker ‘080) at claim 10, Chart A-2 (Tucker ‘556) at claim 10, and Chart A-3 (Tucker Provisional) at claim 10 as if fully set forth herein. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.

Invalidity Contentions: Solaris Zones

Claim 31

’814 Patent Claim 31	Disclosure
31[p] A computing system for performing a plurality of tasks each comprising a plurality of processes comprising:	Solaris Zones, as evidenced by the example citations below, discloses a computing system for performing a plurality of tasks each comprising a plurality of processes. This element is satisfied by the disclosure set forth at claim elements 1[p][1]-[5]. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][1] a system having a plurality of secure containers of associated files accessible to, and for execution on, one or more servers,	Solaris Zones, as evidenced by the example citations below, discloses a system having a plurality of secure containers of associated files accessible to, and for execution on, one or more servers. This element is satisfied by the disclosure set forth at claim elements 1[p][1]-[5]. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][2] each container being mutually exclusive of the other, such that read/write files within a container cannot be shared with other containers,	Solaris Zones, as evidenced by the example citations below, discloses each container being mutually exclusive of the other, such that read/write files within a container cannot be shared with other containers. This element is satisfied by the disclosure set forth at claim elements 1[p][2]-[4], 1[a][6]. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][3] each container of files is said to have its own unique identity associated therewith, said identity comprising at least one of an IP address, a host name, and a Mac_address;	Solaris Zones, as evidenced by the example citations below, discloses each container of files is said to have its own unique identity associated therewith, said identity comprising at least one of an IP address, a host name, and a Mac_address. This element is satisfied by the disclosure set forth at claim 6.

’814 Patent Claim 31	Disclosure
	To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][4] wherein, the plurality of files within each of the plurality of containers comprise one or more application programs including one or more processes, and	Solaris Zones, as evidenced by the example citations below, discloses the plurality of files within each of the plurality of containers comprise one or more application programs including one or more processes. *See, e.g.*, the disclosures set forth at claim element 1[p][3]-[5], 1[a][2]-[5]. This element is satisfied by the disclosure set forth at claim elements 1[p][3]-[5], 1[a][2]-[5]. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][5] associated system files for use in executing the one or more processes wherein the associated system files are files that are copies of files or modified copies of files that remain as part of the operating system,	Solaris Zones, as evidenced by the example citations below, discloses associated system files for use in executing the one or more processes wherein the associated system files are files that are copies of files or modified copies of files that remain as part of the operating system. *See, e.g.*, the disclosures set forth at claim element 1[p][2], 1[a][5]. This element is satisfied by the disclosure set forth at claim elements 1[p][2], 1[a][5]. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][6] each container having its own execution file associated therewith for starting one or more applications, in operation,	Solaris Zones, as evidenced by the example citations below, discloses each container having its own execution file associated therewith for starting one or more applications, in operation. *See, e.g.*, the disclosures set forth at claim element 1[p][3],[5], 1[a][2],[6], 2. This element is satisfied by the disclosure set forth at claim elements 1[p][3],[5], 1[a][2],[6], 2.

’814 Patent Claim 31	Disclosure
	To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][7] each container utilizing a kernel resident on the server and wherein each container exclusively uses a kernel in an underlying operation system in which it is running and is absent its own kernel; and,	Solaris Zones, as evidenced by the example citations below, discloses each container utilizing a kernel resident on the server and wherein each container exclusively uses a kernel in an underlying operation system in which it is running and is absent its own kernel. *See, e.g.*, the disclosures set forth at claim element 1[p][2], 1[a][2]-[3]. This element is satisfied by the disclosure set forth at claim elements 1[p][2], 1[a][2]-[3]. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.
31[a][8] a run time module for monitoring system calls from applications associated with one or more containers and for providing control of the one or more applications.	Solaris Zones, as evidenced by the example citations below, discloses a run time module for monitoring system calls from applications associated with one or more containers and for providing control of the one or more applications. *See, e.g.*, the disclosures set forth at claim element 1[p][2],[4],[5], 1[a][2]. This element is satisfied by the disclosure set forth at claim elements 1[p][2],[4],[5], 1[a][2]. To the extent Plaintiff contends that this reference does not disclose this limitation, the limitation would have been obvious to a person of ordinary skill in the art based on this reference either alone or in combination with (1) the general knowledge of one of ordinary skill in the art, and/or (2) the teachings of the prior art references set forth in Defendant’s other invalidity charts for the ’814 patent with respect to this limitation, as described in Defendant’s Invalidity Contentions.